Self-Proliferating Musical Objects
Michael Longton
School of Music
University of Victoria
Victoria, B.C., Canada

Abstract
A system is described for the automatic generation of musical objects in real time. At the core of the system is the concept of familial interstice.

1. Background
The problem of asserting real-time control over the musical macrostructure turns out to be an exercise in the seemingly impossible to afford the single computer-performer control over a range and complexity of events well in excess of human information-processing bandwidth. More specifically, it is a problem of populating the musical space with a multitude of events while maintaining some degree of rational control over the process. Past efforts to solve it were based on the cloning of objects with some parameter(s) changed (as in simple transposition of a motif), or the stochastic generation of new material, usually with the goal of producing fields of events which grow into families by means of parametric identity.

2. The Game of Life
The particular approach taken here was inspired in large part by John Horton Conway's game of Life, first developed in 1970. Life is a game which is played by a computer without human intervention: the "player" is responsible only for specifying the initial state of the Life "universe" and may even abstain at the outset by specifying a random selection of elements. The Life screen consists of a matrix whose cells are either on or off. Once set in motion, it evolves by means of a very simple set of rules essentially, the state of any cell in the next generation is determined by counting the number of "live" neighbors. The peculiar charm of Life lies in the fact that, even given a random selection of initial live cells, the Life screen can evolve into a field of highly organized patterns which are capable of potentially adjusting or evading. These life forms include "Bentheves," "Tubs," "Pentips," "Tonsels," "Giders," "Spaceships," "Spaceship factories," "Traffic Lights," "Homes," "Tarms," "Snakes," "Lumps of Muck," and a host of others. Though the evolution of Life is totally deterministic, there is no intuitive or mathematical way to predict what form that evolution will take. An entertaining exposition of Life is given in Poundstone (1985).

2.1. Music and Life
The author and several of his colleagues have attempted over a number of years to develop a "musical Life," that is, to map Life screens directly onto sets of musical parameters. Such a simplistic approach seems to be doomed. There is no real parallel between the behavior of Life objects and musical objects and therefore no mapping operation that would be universally applicable. Further, even the three-dimensional space occupied by Life (evolution of the two-dimensional screen over time would be too limiting for most musical purposes.)

The system we are about to describe owes at least as much to generic models of it as it does to Life, but it does attempt to preserve one of the most appealing features of Life: the creation of seemingly "live" objects through the interaction of an object with its neighbors.

3. Self-Proliferating Objects
The solution suggested here owes a good deal to the author's earlier work with Markovian methods of control over the musical macrostructure. The present implementation is rudimentary: it defines the musical object as a data structure containing: (1) a start time (2) a duration, (3) a (starting) frequency, (4) a symbolic instrument label (timbral) and an arbitrary number of user-assignable parameters. Coupled with this set of specificity musical parameters (5) the list of all objects in the current pool
t transfers to the start of the new pool. Each object generates, with the help of its neighbor (the object immediately preceding it) one child, which inherits, unchanged, the parent's role. The child's role

A-5
ICMC 86 Proceedings
but, however, may be inherited from either the parent or the neighbor or from both by means of the set operations of intersection, union, and complement. In the present implementation only the corresponding bit-wise boolean operators (AND, OR, XOR) are implemented, and therefore one can operate only on numerical and not symbolic data. The generation of a simple "child" object is shown below.

```
sub p: [7 10. 400. 480.]
sub f: [XOR, P, N]
```

```
parent p: [2 10. 600. 480.]
parent f: [AND, N, XOR, P]
```

```
child p: [4 6. 612. 480.]
child f: [AND, N, XOR, P]
```

The process is, of course, recursive, with each child generating exactly one offspring.

4. Problems and Peculiarities

A few peculiarities of this approach should be evident. There will normally be several distinct families of objects populating the musical space and objects with pre-defined "new" characteristics will be generated by the interaction of objects from different families. Because of the way the boolean operators behave, incremental generation will tend either to "freeze" parameter values or to reduce them to zero. There will be occasions, however, when it is necessary to produce clones of an object with only one or two parameters changed; for this reason incremental generation is permitted, but as a special case. There is a flag attached to each family's P-net (actually the concept of "family" and P-net are not distinguishable) which specifies whether the neighbor selected will be the immediate predecessor of the same or of a different family.

Negative scheduling is not a problem, since start times are relative to the preceding object and not to time zero. When, as a result of t-net operations, any parameter is reduced to zero, the object is silent. In fact there may be, at any time, a large number of silent objects inhabiting the space. These are all capable of generating audible property, once the null parameters cease to be null (through OR or XOR operations).

The process by which the musical space is populated is hardly intuitive, but it does provide for a high degree of macrostructural control. This is achieved by allowing all current t-sets to be called up and modeled in real time. Since changes to the t-sets are inherited by all descendants, quite drastic changes to the texture can be achieved in this way.

5. Future Improvements

The author intends to develop a more sophisticated version of the system with pseudo-synthetic refinement. Obviously a better way is needed to deal with symbolic data. In future versions the process will be generalized so that P-nets contain only symbolic information, from which separate synthesis routines produce the specific (musical) values.

References